Package ‘HiCExperiment’

May 26, 2026

Title Bioconductor class for interacting with Hi-C files in R

Version 1.13.0

Date 2025-07-22

Description R generic interface to Hi-C contact matrices in `.(m)cool`, `.hic` or HiC-Pro derived formats, as well as other Hi-C processed file formats. Contact matrices can be partially parsed using a random access method, allowing a memory-efficient representation of Hi-C data in R. The `HiCExperiment` class stores the Hi-C contacts parsed from local contact matrix files. `HiCExperiment` instances can be further investigated in R using the `HiContacts` analysis package.

License MIT + file LICENSE

URL https://github.com/js2264/HiCExperiment

BugReports https://github.com/js2264/HiCExperiment/issues

Depends R (>= 4.2)

Imports InteractionSet, strawr, Seqinfo, GenomicRanges, IRanges, S4Vectors, BiocGenerics, BiocIO, BiocParallel, methods, rhdf5, Matrix, vroom, dplyr, stats

Suggests HiContacts, HiContactsData, BiocFileCache, rtracklayer, testthat (>= 3.0.0), BiocStyle, knitr, rmarkdown

biocViews HiC, DNA3DStructure, DataImport

Encoding UTF-8

VignetteBuilder knitr

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/testthat/edition 3

Collate 'HiCExperiment-class.R' 'AggrHiCExperiment-class.R' 'AllGenerics.R' 'HiCExperiment-methods.R' 'AggrHiCExperiment-methods.R' 'PairsFile-class.R' 'ContactsFile-class.R' 'ContactsFile-methods.R' 'CoolFile-class.R' 'CoolFile-methods.R' 'HicFile-class.R' 'HicFile-methods.R' 'HicproFile-class.R' 'HicproFile-methods.R' 'PairsFile-methods.R' 'import-methods.R' 'available.R' 'bin.R' 'checks.R' 'coerce.R' 'data.R' 'export-methods.R' 'globals.R' 'parse-cool.R' 'parse-hic.R' 'parse-hicpro.R' 'parse-pairs.R' 'parse-slices.R' 'reexports.R' 'utils.R' 'zzz.R'

git_url https://git.bioconductor.org/packages/HiCExperiment

git_branch devel

git_last_commit ec0ab68

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-25

Author Jacques Serizay [aut, cre] (ORCID: <`https://orcid.org/0000-0002-4295-0624`>)

Maintainer Jacques Serizay <`jacquesserizay@gmail.com`>

Contents

`.pairs2gi` *Pairs parsing functions*

Description

Pairs parsing functions

Usage

```
.pairs2gi(
  file,
  chr1.field = NULL,
  start1.field = NULL,
  chr2.field = NULL,
  start2.field = NULL,
  strand1.field = NULL,
  strand2.field = NULL,
  frag1.field = NULL,
  frag2.field = NULL,
  nThread = 1,
  nrows = Inf
)
```

Arguments

`file`	pairs file. Default formatting is `<readname>\t<chr1>\t<start1>\t<chr2>\t<start2>`.
`chr1.field`, `start1.field`, `chr2.field`, `start2.field`, `strand1.field`, `strand2.field`, `frag1.field`, `frag2.field`	Index of the column in which each field is contained in the pairs file.
`nThread`	Number of CPUs to use to import the `pairs` file in R
`nrows`	Number of pairs to import

Value

a GInteractions object

`AggrHiCExperiment` `AggrHiCExperiment` *S4 class*

Description

The `AggrHiCExperiment` extends `HiCExperiment` class.

Usage

```
AggrHiCExperiment(
  file,
  resolution = NULL,
  targets,
  flankingBins = 50,
  metadata = list(),
  topologicalFeatures = S4Vectors::SimpleList(),
  pairsFile = NULL,
  bed = NULL,
  maxDistance = NULL,
  BPPARAM = BiocParallel::bpparam()
)
```

```
## S4 method for signature 'AggrHiCExperiment,missing'
slices(x)

## S4 method for signature 'AggrHiCExperiment,character'
slices(x, name)

## S4 method for signature 'AggrHiCExperiment,numeric'
slices(x, name)

## S4 method for signature 'AggrHiCExperiment'
show(object)
```

Arguments

`file`	CoolFile or plain path to a Hi-C contact file
`resolution`	Resolution to use with the Hi-C contact file
`targets`	Set of chromosome coordinates for which interaction counts are extracted from the Hi-C contact file, provided as a GRanges object (for diagnoal-centered loci) or as a GInteractions object (for off-diagonal coordinates).
`flankingBins`	Number of bins on each flank of the bins containing input targets.
`metadata`	list of metadata
`topologicalFeatures`	topologicalFeatures provided as a named SimpleList
`pairsFile`	Path to an associated .pairs file
`bed`	Path to regions file generated by HiC-Pro
`maxDistance`	Maximum distance to use when compiling distance decay
`BPPARAM`	BiocParallel parameters
`x, object`	A `AggrHiCExperiment` object.
`name`	The name/index of slices to extract.

Value

An `AggrHiCExperiment` object.

Slots

- `fileName` Path of Hi-C contact file
- `resolutions` Resolutions available in the Hi-C contact file.
- `resolution` Current resolution
- `interactions` Genomic Interactions extracted from the Hi-C contact file
- `scores` Available interaction scores.
- `slices` Available interaction slices.
- `topologicalFeatures` Topological features associated with the dataset (e.g. loops (\<Pairs\>), borders (\<GRanges\>), viewpoints (\<GRanges\>), etc...)
- `pairsFile` Path to the .pairs file associated with the Hi-C contact file
- `metadata` metadata associated with the Hi-C contact file.

See Also

`HiCExperiment()`

Examples

```
fpath <- HiContactsData::HiContactsData('yeast_wt', 'mcool')
data(centros_yeast)
x <- AggrHiCExperiment(
  file = fpath,
  resolution = 8000,
  targets = centros_yeast[c(4, 7)]
)
x
slices(x, 'count')[1:10, 1:10, 1]
```

`AllGenerics` *Generic functions*

Description

Generics functions created in HiCExperiment package.

Arguments

`x`	Passed to corresponding method
`name`	Passed to corresponding method
`value`	Passed to corresponding method
`...`	Passed to corresponding method

`as` *Coercing functions*

Description

Coercing functions available for HiCExperiment objects.

Usage

```
## S4 method for signature 'HiCExperiment'
as.matrix(x, use.scores = "balanced", sparse = FALSE)

## S4 method for signature 'HiCExperiment'
as.data.frame(x)

gi2cm(gi, use.scores = "score")

cm2matrix(cm, replace_NA = NA, sparse = FALSE)

df2gi(
```

```
  df,
  seqnames1 = "seqnames1",
  start1 = "start1",
  end1 = "end1",
  seqnames2 = "seqnames2",
  start2 = "start2",
  end2 = "end2"
)
```

Arguments

`x`	HiCExperiment object
`use.scores`	Which scores to use to inflate GInteractions
`sparse`	Whether to return the contact matrix as a sparse matrix
`gi`	GInteractions object
`cm`	A `ContactMatrix` object
`replace_NA`	Replace NA values
`df`	A `data.frame` object
`seqnames1`, `start1`, `end1`, `seqnames2`, `start2`, `end2`	Names (as strings) of columns containing corresponding information in a data.frame parsed into GInteractions (default: FALSE)

Examples

```
mcoolPath <- HiContactsData::HiContactsData('yeast_wt', 'mcool')
contacts <- import(mcoolPath, focus = 'XVI', resolution = 16000, format = 'cool')
gis <- interactions(contacts)
cm <- gi2cm(gis, 'balanced')
cm
cm2matrix(cm)[1:10, 1:10]
df2gi(data.frame(
    chr1 = 'I', start1 = 10, end1 = 100,
    chr2 = 'I', start2 = 40, end2 = 1000,
    score = 12,
    weight = 0.234,
    filtered = TRUE
), seqnames1 = 'chr1', seqnames2 = 'chr2')
```

`bin-methods` *HiCExperiment binning methods*

Description

HiCExperiment binning methods

Usage

```
## S4 method for signature 'GInteractions,numeric'
bin(x, resolution, seqinfo = NULL)

## S4 method for signature 'PairsFile,numeric'
bin(x, resolution, seqinfo = NULL)
```

Arguments

`x`	A PairsFile or GInteractions object
`resolution`	Which resolution to use to bin the interactions
`seqinfo`	Seqinfo object

Examples

```
pairsf <- HiContactsData::HiContactsData('yeast_wt', 'pairs.gz')
pf <- PairsFile(pairsf)
```

---

checks *Checks functions*

Description

Internal functions to validate the nature/structure of (m)cool files or `HiCExperiment` objects. All these check functions should return a logical.

Usage

```
.check_cool_file(path)

.check_cool_format(path, resolution, ...)

.is_mcool(path)

.is_cool(path)

.check_hic_file(path)

.check_hic_format(path, resolution, ...)

.is_hic(path)

.check_hicpro_files(path, bed)

.is_hicpro_matrix(path)

.is_hicpro_regions(bed)

.check_resolution(contacts, resolution)

.check_scores(contacts, use.scores)

.is_square(pair)
```

Arguments

`path`	Path of a (m)cool file
`resolution`	Resolution
`...`	`HiCExperiment` object, arguments passed on by other functions
`bed`	Path to regions file generated by HiC-Pro
`contacts`	A `HiCExperiment` object
`use.scores`	Name of scores to check
`pair`	Pairs object with length of 1

Value

Logical

`ContactsFile-class` `ContactsFile` *S4 class*

Description

The `ContactsFile` class describes a `BiocFile` object, pointing to the location of an Hi-C matrix file (cool, mcool, hic, hicpro, ...) and containing additional slots:

1. resolution: at which resolution the associated mcool file should be parsed
2. pairsFile: the path (in plain character) to an optional pairs file (stored as a `PairsFile` object);
3. metadata: a list. If the CoolFile is created by `HiCool`, it will contain two elements: `log` (path to `HiCool` processing log file) and `stats` (aggregating some stats from `HiCool` mapping).

ContactsFile methods.

Arguments

`path`	String; path to an Hi-C matrix file (cool, mcool, hic, hicpro)
`resolution`	numeric; resolution to use with Hi-C matrix file
`pairsFile`	String; path to a pairs file
`metadata`	list.
`object`	A `ContactsFile` object.
`x`	A `ContactsFile` object.

Slots

`resolution` numeric value or NULL

`pairsFile` PairsFile object

`metadata` list

See Also

`CoolFile()`, `HicFile()`, `HicproFile()`

`CoolFile-class` `CoolFile` *S4 class*

Description

The `CoolFile` class describes a `BiocFile` object, pointing to the location of an Hi-C matrix file (cool, mcool, hic, hicpro, ...) and containing additional slots:

1. resolution: at which resolution the associated mcool file should be parsed
2. pairsFile: the path (in plain character) to an optional pairs file (stored as a `PairsFile` object);
3. metadata: a list. If the CoolFile is created by `HiCool`, it will contain two elements: `log` (path to `HiCool` processing log file) and `stats` (aggregating some stats from `HiCool` mapping).

CoolFile methods.

Arguments

`path`	String; path to a (m)cool file
`resolution`	numeric; resolution to use with mcool file
`pairsFile`	String; path to a pairs file
`metadata`	list; if the CoolFile object was generated by HiCool::HiCool, this list contains the path to log file, some statistics regarding the number of pairs obtained by hicstuff as well as the arguments and the hash ID used by HiCool.
`object`	A `CoolFile` object.

See Also

`HicFile()`, `HicproFile()`

Examples

```
mcoolPath <- HiContactsData::HiContactsData('yeast_wt', 'mcool')
pairsPath <- HiContactsData::HiContactsData('yeast_wt', 'pairs.gz')
cf <- CoolFile(
  mcoolPath,
  resolution = 2000,
  pairsFile = pairsPath,
  metadata = list(info = 'Yeast WT Hi-C exp.')
)
cf
resolution(cf)
pairsFile(cf)
metadata(cf)
```

`data` *Example datasets provided in* `HiCExperiment` *&* `HiContactsData`

Description

Example datasets provided in `HiCExperiment` & `HiContactsData`

Usage

```
data(centros_yeast)

contacts_yeast(full = FALSE)

contacts_yeast_eco1(full = FALSE)
```

Arguments

`full`	Whether to import all interactions

Format

An object of class `"GRanges"`.

Source

HiContacts

Examples

```
data(centros_yeast)
centros_yeast
contacts_yeast()
```

`export-methods` *HiCExperiment export methods*

Description

Export methods to save a HiCExperiment object into a set of HiC-Pro-style files (matrix & regions files)

Usage

```
## S4 method for signature 'HiCExperiment,missing,character'
export(object, prefix, format, ...)
```

Arguments

`object`	A HiCExperiment object
`prefix`	Prefix used when generating output file(s).
`format`	File format. Available: `cool` and `HiC-Pro`.
`...`	Extra arguments to use when exporting to `cool`. Can be `metadata <string>` or `chunksize <integer>`.

Value

Path to saved files

Examples

```
##################################################################
## ----------- Importing .(m)cool contact matrices ---------- ##
##################################################################

mcoolPath <- HiContactsData::HiContactsData('yeast_wt', 'mcool')
hic <- import(mcoolPath, format = 'mcool', resolution = 16000)
export(hic["II"], prefix = 'subset_chrII', format = 'cool')
export(hic["II"], prefix = 'subset_chrII', format = 'HiC-Pro')
```

`HiCExperiment` `HiCExperiment` *S4 class*

Description

The `HiCExperiment` class describes Hi-C contact files imported in R, either through the `HiCExperiment` constructor function or using the `import` method implemented by `HiCExperiment` package.

Usage

```
HiCExperiment(
  file,
  resolution = NULL,
  focus = NULL,
  metadata = list(),
 topologicalFeatures = S4Vectors::SimpleList(compartments = GenomicRanges::GRanges(),
    borders = GenomicRanges::GRanges(), loops =
   InteractionSet::GInteractions(GenomicRanges::GRanges(), GenomicRanges::GRanges()),
    viewpoints = GenomicRanges::GRanges()),
  pairsFile = NULL,
  bed = NULL
)

makeHiCExperimentFromGInteractions(gi)

## S4 method for signature 'HiCExperiment'
resolutions(x)

## S4 method for signature 'HiCExperiment'
```

```
resolution(x)

## S4 method for signature 'HiCExperiment'
focus(x)

## S4 replacement method for signature 'HiCExperiment,character'
focus(x) <- value

## S4 method for signature 'HiCExperiment,numeric'
zoom(x, resolution)

## S4 method for signature 'HiCExperiment,character'
refocus(x, focus)

## S4 method for signature 'HiCExperiment,missing'
scores(x)

## S4 method for signature 'HiCExperiment,character'
scores(x, name)

## S4 method for signature 'HiCExperiment,numeric'
scores(x, name)

## S4 replacement method for signature 'HiCExperiment,character,numeric'
scores(x, name) <- value

## S4 method for signature 'HiCExperiment,missing'
topologicalFeatures(x)

## S4 method for signature 'HiCExperiment,character'
topologicalFeatures(x, name)

## S4 method for signature 'HiCExperiment,numeric'
topologicalFeatures(x, name)

## S4 replacement method for signature 'HiCExperiment,character,GRangesOrGInteractions'
topologicalFeatures(x, name) <- value

## S4 method for signature 'HiCExperiment'
pairsFile(x)

## S4 replacement method for signature 'HiCExperiment,character'
pairsFile(x) <- value

## S4 replacement method for signature 'HiCExperiment,list'
metadata(x) <- value

## S4 method for signature 'HiCExperiment,numeric'
subsetByOverlaps(x, ranges)

## S4 method for signature 'HiCExperiment,logical'
subsetByOverlaps(x, ranges)
```

```
## S4 method for signature 'HiCExperiment,GRanges'
subsetByOverlaps(x, ranges, type = c("within", "any"))

## S4 method for signature 'HiCExperiment,GInteractions'
subsetByOverlaps(x, ranges)

## S4 method for signature 'HiCExperiment,Pairs'
subsetByOverlaps(x, ranges)

## S4 method for signature 'HiCExperiment,numeric,ANY,ANY'
x[i]

## S4 method for signature 'HiCExperiment,GRanges,ANY,ANY'
x[i]

## S4 method for signature 'HiCExperiment,logical,ANY,ANY'
x[i]

## S4 method for signature 'HiCExperiment,GInteractions,ANY,ANY'
x[i]

## S4 method for signature 'HiCExperiment,Pairs,ANY,ANY'
x[i]

## S4 method for signature 'HiCExperiment,character,ANY,ANY'
x[i]

## S4 method for signature 'HiCExperiment'
fileName(object)

## S4 method for signature 'HiCExperiment'
interactions(x, fillout.regions = FALSE)

## S4 replacement method for signature 'HiCExperiment,GInteractions'
interactions(x) <- value

## S4 method for signature 'HiCExperiment'
length(x)

## S4 replacement method for signature 'HiCExperiment'
x$name <- value

## S4 method for signature 'HiCExperiment'
x$name

## S4 method for signature 'HiCExperiment'
seqinfo(x)

## S4 method for signature 'HiCExperiment'
bins(x)
```

```
## S4 method for signature 'HiCExperiment'
anchors(x)

## S4 method for signature 'HiCExperiment'
regions(x)

## S4 method for signature 'HiCExperiment'
cis(x)

## S4 method for signature 'HiCExperiment'
trans(x)
```

Arguments

`file`	CoolFile or plain path to a Hi-C contact file
`resolution`	Resolution to use with the Hi-C contact file
`focus`	Chromosome coordinates for which interaction counts are extracted from the Hi-C contact file, provided as a character string (e.g. "II:4001-5000"). If not provided, the entire Hi-C contact file will be imported.
`metadata`	list of metadata
`topologicalFeatures`	topologicalFeatures provided as a named SimpleList
`pairsFile`	Path to an associated .pairs file (optional)
`bed`	Path to regions file generated by HiC-Pro (optional)
`gi`	GInteractions object
`x`	A `HiCExperiment` object.
`value`	Value to add to topologicalFeatures, scores, pairsFile or metadata slots.
`name`	Name of the element to access in topologicalFeatures or scores SimpleLists.
`type`	any of `within` or `any`, to subset interactions by overlap with a provided GRanges.
`i, ranges`	a GRanges, coordinates in character, or boolean vector to subset a HiCExperiment
`object`	A `HiCExperiment` object.
`fillout.regions`	Whehter to add missing regions to GInteractions' regions?

Value

An `HiCExperiment` object.

Slots

`fileName` Path of Hi-C contact file

`focus` Chr. coordinates for which interaction counts are extracted from the Hi-C contact file.

`resolutions` Resolutions available in the Hi-C contact file.

`resolution` Current resolution

`interactions` Genomic Interactions extracted from the Hi-C contact file

`scores` Available interaction scores.

`topologicalFeatures` Topological features associated with the dataset (e.g. loops (\<GInteractions\>), borders (\<GRanges\>), viewpoints (\<GRanges\>), etc...)

`pairsFile` Path to the .pairs file associated with the Hi-C contact file

`metadata` metadata associated with the Hi-C contact file.

See Also

`AggrHiCExperiment()`, `CoolFile()`, `HicFile()`, `HicproFile()`, `PairsFile()`

Examples

```
######################################################################
## Create a HiCExperiment object from a disk-stored contact matrix ##
######################################################################

mcool_file <- HiContactsData::HiContactsData("yeast_wt", "mcool")
pairs_file <- HiContactsData::HiContactsData("yeast_wt", "pairs.gz")
contacts <- HiCExperiment(
    file = mcool_file,
    resolution = 8000L,
    pairsFile = pairs_file
)
contacts

######################################################################
## ----- Manually create a HiCExperiment from GInteractions ------ ##
######################################################################

gis <- interactions(contacts)[1:1000]
contacts2 <- makeHiCExperimentFromGInteractions(gis)
contacts2

######################################################################
## -------- Slots present in an HiCExperiment object ------------- ##
######################################################################

fileName(contacts)
focus(contacts)
resolutions(contacts)
resolution(contacts)
interactions(contacts)
scores(contacts)
topologicalFeatures(contacts)
pairsFile(contacts)

######################################################################
## ---------------------- Slot getters ---------------------------- ##
######################################################################

scores(contacts, 1) |> head()
scores(contacts, 'balanced') |> head()
topologicalFeatures(contacts, 1)

######################################################################
## ---------------------- Slot setters ---------------------------- ##
######################################################################
```

```
scores(contacts, 'random') <- runif(length(contacts))
topologicalFeatures(contacts, 'loops') <- InteractionSet::GInteractions(
  GenomicRanges::GRanges('II:15324'),
  GenomicRanges::GRanges('II:24310')
)
pairsFile(contacts) <- HiContactsData('yeast_wt', 'pairs.gz')

#####################################################################
## ------------------ Subsetting functions ----------------------- ##
#####################################################################

contacts[1:100]
contacts['II']
contacts[c('II', 'III')]
contacts['II|III']
contacts['II:10001-30000|III:50001-90000']

#####################################################################
## --------------------- Utils functions ------------------------- ##
#####################################################################
## Adapted from other packages

seqinfo(contacts)
bins(contacts)
anchors(contacts)
regions(contacts)

#####################################################################
## ------------- Coercing HiCExperiment objects ------------------ ##
#####################################################################

as(contacts, 'GInteractions')
as(contacts, 'ContactMatrix')
as(contacts, 'matrix')[seq_len(10), seq_len(10)]
as(contacts, 'data.frame')[seq_len(10), seq_len(10)]
```

`HiCExperiment utils` *Utils functions*

Description

Utilities to facilitate parsing/handling of coordinates, GInteractions, Pairs, ... These functions are not exported.

Usage

```
splitCoords(coords)

coords2char(coords, big.mark = ",")

char2coords(char)

sortPairs(pairs)
```

```
asGInteractions(df)

sdiag(A, k = 0)

.df2symmmat(diag, score)

distanceDecay(dump, threshold = NULL)

detrendingModel(file, resolution)

.fixRegions(gis, bins, coords)
```

Arguments

`coords`	A set of genomic coordinates (either as a GRanges object or as a character string)
`big.mark`	Separator for thousands when printing out genomic coordinates as character
`char`	char (e.g. "II:30001-50000" or "II:30001-50000|II:60001-80000")
`pairs`	Pairs object
`df`	a data.frame to turn into a GInteraction object.
`A`	Numerical matrix
`k`	secondary diagonal k
`diag`	vector of distances to diagonal
`score`	scores to parse into symmetrical matrix
`dump`	dumped contacts as GInteractions, e.g. from .dumpCool
`threshold`	maximum distance to compute distance decay for
`file`	path to a HiC contact matrix file
`resolution`	Resolution to use with the HiC contact matrix file
`gis`	GInteractions object
`bins`	Larger set of regions (usually bins from HiCExperiment)

Value

Reformatted coordinates or GInteractions.

`HicFile-class` `HicFile` *S4 class*

Description

The `HicFile` class describes a `BiocFile` object, pointing to the location of a .hic file (usually created with juicer) and containing 3 additional slots:

1. resolution: at which resolution the associated .hic file should be parsed;
2. pairsFile: the path (in plain character) to an optional pairs file (stored as a `PairsFile` object);
3. metadata: a list metadata

HicFile methods.

Arguments

`path`	String; path to a .hic file
`resolution`	numeric; resolution to use with mcool file
`pairsFile`	String; path to a pairs file
`metadata`	list.
`object`	A `HicFile` object.

See Also

`CoolFile()`, `HicproFile()`

Examples

```
hicPath <- HiContactsData::HiContactsData('yeast_wt', 'hic')
pairsPath <- HiContactsData::HiContactsData('yeast_wt', 'pairs.gz')
hic <- HicFile(
  hicPath,
  resolution = 16000,
  pairsFile = pairsPath,
  metadata = list(type = 'example')
)
hic
resolution(hic)
pairsFile(hic)
metadata(hic)
```

`HicproFile-class` `HicproFile` *S4 class*

Description

The `HicproFile` class describes a `BiocFile` object, pointing to the location of a HiC-Pro-generated matrix file and containing 4 additional slots:

1. bed: path to the matching .bed file generated by HiC-Pro;
2. resolution: at which resolution the associated mcool file should be parsed ;
3. pairsFile: the path (in plain character) to an optional pairs file (stored as a `PairsFile` object);
4. metadata: a list metadata

HicproFile methods.

Arguments

`path`	String; path to the HiC-Pro output .matrix file (matrix file)
`bed`	String; path to the HiC-Pro output .bed file (regions file)
`pairsFile`	String; path to a pairs file
`metadata`	list.
`object`	A `HicproFile` object.

Slots

bed Path to the matching `.bed` file generated by HiC-Pro

See Also

`CoolFile()`, `HicFile()`

Examples

```
hicproMatrixPath <- HiContactsData::HiContactsData('yeast_wt', 'hicpro_matrix')
hicproBedPath <- HiContactsData::HiContactsData('yeast_wt', 'hicpro_bed')
pairsPath <- HiContactsData::HiContactsData('yeast_wt', 'pairs.gz')
hicpro <- HicproFile(
  hicproMatrixPath, bed = hicproBedPath, pairs = pairsPath ,
  metadata = list(type = 'example')
)
hicpro
resolution(hicpro)
pairsFile(hicpro)
metadata(hicpro)
```

`import-methods` *HiCExperiment import methods*

Description

Import methods to parse Hi-C files (`.(m)cool`, `.hic`, HiC-Pro derived matrices, pairs files) into data structures implemented in the HiCExperiment package.

Usage

```
import(con, format, text, ...)

## S4 method for signature 'ANY'
availableResolutions(x, ...)

## S4 method for signature 'CoolFile'
availableResolutions(x)

## S4 method for signature 'HicFile'
availableResolutions(x)

## S4 method for signature 'HicproFile'
availableResolutions(x)

## S4 method for signature 'ANY'
availableChromosomes(x, ...)

## S4 method for signature 'CoolFile'
availableChromosomes(x)

## S4 method for signature 'HicFile'
```

```
availableChromosomes(x)

## S4 method for signature 'HicproFile'
availableChromosomes(x)
```

Arguments

`...`	Extra parameters to pass to format-specific methods. A list of possible arguments is provided in the next section.
`con, x`	Path or connection to a cool, mcool, .hic or HiC-Pro derived files. Can also be a path to a pairs file.
`format`	The format of the output. If missing and 'con' is a filename, the format is derived from the file extension. This argument is unnecessary when files are directly provided as `CoolFile`, `HicFile`, `HicproFile` or `PairsFile`.
`text`	If 'con' is missing, this can be a character vector directly providing the string data to import.

Value

A `HiCExperiment` or `GInteractions` object

import arguments for ContactFile class

`ContactFile` class gathers `CoolFile`, `HicFile` and `HicproFile` classes. When importing a `ContactFile` object in R, two main arguments can be provided besides the `ContactFile` itself:

- `resolution`: Resolutions available in the disk-stored contact matrix can be listed using `availableResolutions(f`
- `focus`: A genomic locus (or pair of loci) provided as a string. It can be any of the following string structures:
    - "`II`" or "`II:20001-30000`": this will extract a symmetrical square HiCExperiment object, of an entire chromosome or an portion of it.
    - "`II|III`" or "`II:20001-30000|III:40001-90000`": this will extract a non-symmetrical HiCExperiment object, with an entire or portion of different chromosomes on each axis.

Examples

```
#################################################################
## ----------- Importing .(m)cool contact matrices ---------- ##
#################################################################

mcoolPath <- HiContactsData::HiContactsData('yeast_wt', 'mcool')
availableResolutions(mcoolPath)
availableChromosomes(mcoolPath)
import(mcoolPath, resolution = 16000, focus = 'XVI', format = 'cool')

#################################################################
## ------------ Importing .hic contact matrices ------------- ##
#################################################################

hicPath <- HiContactsData::HiContactsData('yeast_wt', 'hic')
availableResolutions(hicPath)
availableChromosomes(hicPath)
import(hicPath, resolution = 16000, focus = 'XVI', format = 'hic')
```

```
################################################################
## ------- Importing HiC-Pro derived contact matrices ------- ##
################################################################

hicproMatrixPath <- HiContactsData::HiContactsData('yeast_wt', 'hicpro_matrix')
hicproBedPath <- HiContactsData::HiContactsData('yeast_wt', 'hicpro_bed')
availableResolutions(hicproMatrixPath, hicproBedPath)
availableChromosomes(hicproMatrixPath, hicproBedPath)
import(hicproMatrixPath, bed = hicproBedPath, format = 'hicpro')
```

`multi2Query` *Querying multiple slices of a contact matrix*

Description

These functions are the workhorse internal functions used to extract counts from multiple genomic coordinates in a Hi-C contact matrix.

Usage

```
.multi2DQuery(
  file,
  resolution,
  pairs,
  maxDistance = NULL,
  bed = NULL,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

`file`	path to a Hi-C contact file (can be any format, (m)cool, .hic, or HiC-Pro-derived)
`resolution`	resolution to use to import matrix over specified targets
`pairs`	slices to read, provided as a Pairs object
`maxDistance`	Maximum distance to use when compiling distance decay
`bed`	associated bed file for HiC-Pro derived contact matrix.
`BPPARAM`	BiocParallel parameters

Value

a GInteractions object with `count`, `balanced`, `detrended` and `expected` scores

`PairsFile-class` `PairsFile` *S4 class*

Description

The `PairsFile` class describes a `BiocFile` object, pointing to the location of pairs file, typically generated by `HiCool::HiCool()`.

PairsFile methods

Arguments

`x`	Path to a pairs file

See Also

`CoolFile()`, `HicFile()`, `HicproFile()`

Examples

```
pairsPath <- HiContactsData::HiContactsData('yeast_wt', 'pairs.gz')
pf <- PairsFile(pairsPath)
pf
pairsFile(pf)
```

`parse-cool` *Parsing (m)cool files*

Description

These functions are the workhorse internal functions used to import a `.(m)cool` file as GInteractions (wrapped into a `HiCExperiment` object by `HiCExperiment()` function).

Usage

```
.getCoolAnchors(file, resolution = NULL, balanced = "cooler")

.getCountsFromPair(file, pair, anchors, resolution = NULL)

.getCounts(file, coords, anchors, resolution = NULL)

.fetchCool(file, path, resolution = NULL, idx = NULL, ...)

.dumpCool(file, resolution = NULL)

.lsCoolFiles(file, verbose = FALSE)

.lsCoolResolutions(file, verbose = FALSE)

.cool2seqinfo(file, resolution = NULL)

.cool2gi(file, coords = NULL, resolution = NULL)
```

Arguments

`file`	path to a Hi-C contact file (in (m)cool format)
`resolution`	resolution of the contact matrix
`balanced`	import balancing scores
`pair`	Genomic coordinates to extract contacts for, stored as a Pairs of GRanges (e.g. S4Vectors::Pairs(GRanges("II:200000-300000"), GRanges("II:70000-100000"))).
`anchors`	anchors from .getCoolAnchors()
`coords`	Genomic coordinates to extract contacts for, stored as a GRanges object
`path`	Internal path of the cool file to check
`idx`	Index to extract from the cool (HDF5) file
`...`	Other arguments passed to .fetchCool
`verbose`	Print resolutions in the console

Value

Silently, a numerical vector of resolutions stored in the cool file

---

`parse-hic` *Parsing hic files*

Description

These functions are the workhorse internal functions used to import a `.hic` file as GInteractions (wrapped into a `HiCExperiment` object by `HiCExperiment()` function).

Usage

```
.hic2gi(file, coords = NULL, resolution = NULL)

.lsHicResolutions(file, verbose = FALSE)

.getHicAnchors(file, resolution = NULL)

.hic2seqinfo(file)

.dumpHic(file, resolution = NULL)
```

Arguments

`file`	path to a Hi-C contact file in .hic format
`coords`	NULL, character, or GRanges. Can also be a Pairs object of paired GRanges (length of 1).
`resolution`	resolution of the contact matrix to use
`verbose`	Print resolutions in the console

Value

a GInteractions object

vector

`parse-hicpro` *Parsing hicpro files (matrix & bed)*

Description

These functions are the workhorse internal functions used to import HiC-Pro `.matrix` and `.bed` files as GInteractions (wrapped into a `HiCExperiment` object by `HiCExperiment()` function).

Usage

```
.hicpro2gi(file, bed)

.getHicproAnchors(bed)

.hicpro2seqinfo(bed)

.dumpHicpro(file, bed)
```

Arguments

`file`	path to a matrix file from HiC-Pro
`bed`	path to the regions file generated by HiC-Pro

Value

a GInteractions object

`reexports` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

S4Vectors `metadata`

Index